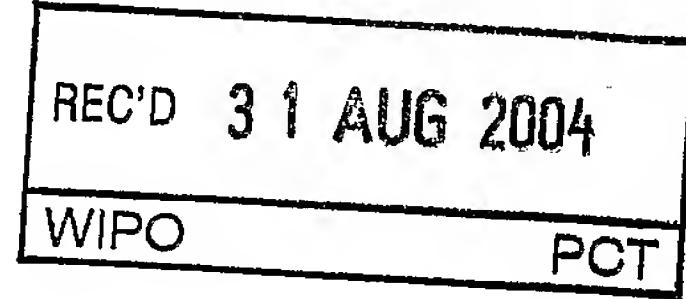


证 明

本证明之附件是向本局提交的下列专利申请副本



申 请 日: 2003. 12. 29

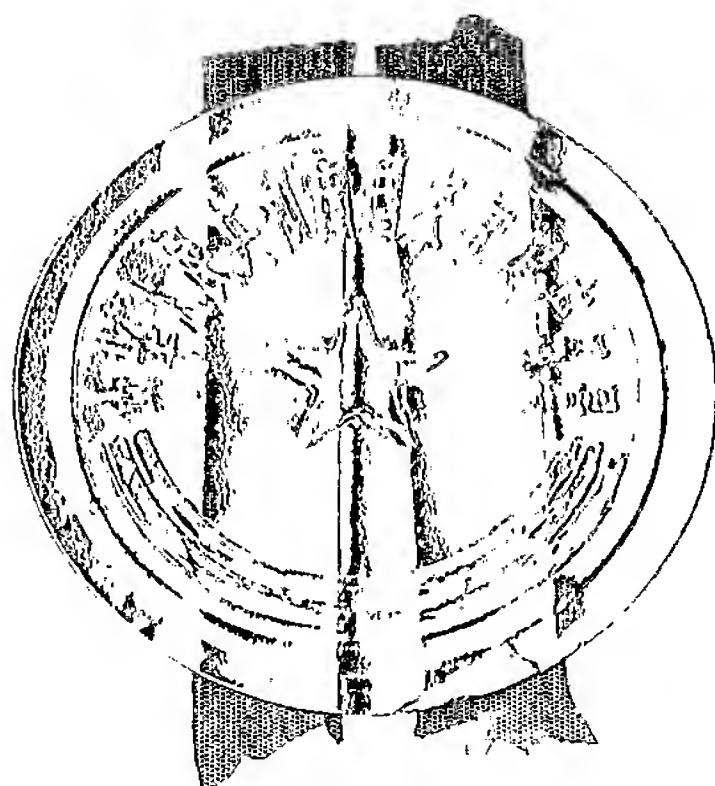
申 请 号: 2003101129721

申 请 类 别: 发明

发明创造名称: 动态循环流水线的交叠式命令提交方法

申 请 人: 中兴通讯股份有限公司

发明人或设计人: 陈虎、任敏



PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

中华人民共和国
国家知识产权局局长

王景川

2004 年 7 月 9 日

1、一种动态循环流水线的交叠式命令提交方法，用于具有流水线结构的芯片中，包括以下步骤：

- 5 (a) 从命令缓冲读取命令；
 (b) 对命令进行译码，判定是否非法指令，如果是，返回步骤 (a)，否则，执行下一步；
 (c) 对命令的操作数进行预处理，准备流水线各段的初始操作参数，存入初始化寄存器；
10 (d) 判断流水线是否处于非满状态，如果是，直接插入新命令并结束，否则等待流水线中命令在退出前的上一流水线周期发出的退出信号；
 (e) 收到所述退出信号后，判断待插入的新命令和将退出的老命令是否存在命令相关性，如果是，在所述老命令退出后再插入新命令，并结束，否则执行下一步；
15 (f) 在所述老命令在流水线的最后一次循环中，向流水线提交新命令。

2、如权利要求 1 所述的命令提交方法，其特征在于，所述退出信号是在所述新命令加入流水线段的前两级发出的。

- 3、如权利要求 1 所述的命令提交方法，其特征在于，所述命令相关性指所述新、老命令不能同时使用同一流水线段上的硬件处理模块。
20 4、如权利要求 1 所述的命令提交方法，其特征在于，在所述步骤 (e) 中，还判断新、老命令应在哪一级流水线段进行现场切换，并在新、老命令交叠的相应流水线段上完成现场切换。

- 25 5、如权利要求 1 所述的命令提交方法，其特征在于，在所述步骤 (e) 中，还判断所述新命令和老命令是否存在现场冲突，如果存在现场冲突，提交时将新命令的现场加入到流水线中，老命令现场则进入现场分支并一直维持到老命令最后一次使用该现场为止；如果不存在现场冲突，提交后在相应的流水线段上完成现场切换。

- 6、如权利要求 1 所述的命令提交方法，其特征在于，所述步骤 (c) 中各类命令都还需要提供进入流水线的初始状态。
30 7、如权利要求 1 所述的命令提交方法，其特征在于，所述命令包括读

000-112-200

6

写存储器命令、读写控制寄存器命令以及多种查找命令。

动态循环流水线的交叠式命令提交方法

5 技术领域

本发明涉及网络通讯等技术领域的各类芯片设计，尤其涉及芯片内环行流水线处理的命令提交方法。

背景技术

10 在通信芯片的设计中，为了满足通信网络速度和容量的快速增长的需求，流水线处理结构已被越多的芯片采用。流水线技术就是计算机技术中同时具备空间并行性和时间并行性的技术，它把一个顺序处理过程分解成若干个子处理过程，每个过程能在专用的独立模块上有效地并发工作。这些子处理过程叫做段，每个流水线段由现场寄存器和硬件处理模块（纯逻辑电路）
15 组成，前者为后者提供输入，后者的输出指向下一级的现场寄存器，在时钟脉冲的作用下，各段将其完成处理的结果同时传送到下一段。及时地向流水线提交命令，保证命令在流水线中不断地流动，是获得高效流水线的关键所在。

图 1 所示是现有技术中环形流水线的命令提交方法，图中的纵坐标表示
20 流水线的级数，共六级流水。横坐标表示时间，A、B、C、D、E、F、G 代表执行的命令。命令的插入位置是流水线的第一级，而命令退出的位置是流水线的第三级。在所有流水线段都满的情况下，A、B、C、D、E、F 都在流水线中运行，当前的嵌入处理器都是在等待流水线命令完全退出流水线循环后，才插入新的命令，如图所示，A 条命令完全退出循环流水线后才插入新的 G 命令，在流水线的第四级到第六级出现了“气泡”。从而无法保证流水
25 线不断的流动，影响了流水线各部件的利用率和命令执行的并行性。

发明内容

本发明要解决的技术问题是提供一种动态循环流水线的交叠式命令提
30 交方法，能避免“气泡”的出现，提高命令执行的并行性和流水线各部件的

利用率。

为了解决上述技术问题，本发明提供了一种动态循环流水线的交叠式命令提交方法，用于具有流水线结构的芯片中，包括以下步骤：

- (a) 从命令缓冲读取命令；
- 5 (b) 对命令进行译码，判定是否非法指令，如果是，返回步骤 (a)，否则，执行下一步；
- (c) 对命令的操作数进行预处理，准备流水线各段的初始操作参数，存入初始化寄存器；
- (d) 判断流水线是否处于非满状态，如果是，直接插入新命令并结束，
10 否则等待流水线中命令在退出前的上一流水线周期发出的退出信号；
- (e) 收到所述退出信号后，判断待插入的新命令和将退出的老命令是否存在命令相关性，如果是，在所述老命令退出后再插入新命令，并结束，否则执行下一步；
- (f) 在所述老命令在流水线的最后一次循环中，向流水线提交新命令。
15

上述方法可具有如下特点：所述退出信号是在所述新命令加入流水线段的前两级发出的。

上述方法可具有如下特点：所述命令相关性指所述新、老命令不能同时使用同一流水线段上的硬件处理模块。

上述方法可具有如下特点：在所述步骤 (e) 中，还判断新、老命令应在哪一级流水线段进行现场切换，并在新、老命令交叠的相应流水线段上完成现场切换。
20

为了解决现场冲突，上述方法可具有如下特点：在所述步骤 (e) 中，还判断所述新命令和老命令是否存在现场冲突，如果存在现场冲突，提交时将新命令的现场加入到流水线中，老命令现场则进入现场分支并一直维持到老命令最后一次使用该现场为止；如果不存在现场冲突，提交后在相应的流水线段上完成现场切换。
25

由上可知，采用本发明的交叠式命令提交方法，可减少命令提交的间隔，提高了流水线执行单元的并行性，从而缩短了命令在芯片中的处理周期，使芯片在单位时间内可处理更多的命令。此外，命令提交部件和流水线执行部件能够分别连续不断的运行，部件的空闲状态减至最低。
30

附图说明

图 1 是现有技术流水线的命令提交方法。

图 2 是本发明实施例芯片 CHIP 的结构框图。

5 图 3 是图 2 中命令解释器的结构框图。

图 4 是本发明实施例命令解释器交叠式提交命令的流程图。

图 5 是本发明实施例的交叠式命令提交方法的示意图。

图 6 是本发明实施例现场切换硬件结构的示意图。

图 7 是本发明实施例现场分支硬件结构的示意图。

10

具体实施方式

下面以一种通信芯片（CHIP）中流水线的命令提交为例，详细说明本发明的实施方法。该芯片的结构如图 2 所示，CHIP 1 中的主机接口 11 从主机接收各种命令如：存储器的读/写访问、控制寄存器的读写访问、不同算法的查找命令等，存入输入缓冲 12，由命令处理单元 13 执行这些命令的操作，当命令执行完毕后，命令的操作结果存放在结果缓冲（result_buffer）14 中，由主机取回。

在本实施例中的 CHIP 支持如下命令：

读写存储器命令；

20 读写控制寄存器命令；以及

查找命令，支持 8 种查找算法。

命令处理单元由命令解释器 131 和流水线执行部件 132 两部分组成。命令解释器负责从输入缓冲中的命令缓冲和数据缓冲中取出命令字和操作数，将其送入流水线。

25 在流水线内部，各命令执行的操作顺序固定，但是命令何时加入流水线执行则由流水线状态决定。各命令操作之间在流水线段上的切换是连续的，但由于可能存在循环操作，进入流水线命令的顺序和执行完毕退出流水线的命令顺序有可能并不一致。命令执行完毕后，我们利用结果缓冲对各命令结果进行有序回收。各命令的递推操作，需要多个流水线周期连续执行，其循环过程也由流水线控制。本实施例的流水线分为六段，可同时容纳六条命令，

每条命令的现场随流水线而逐级流动。

命令解释器和流水线执行部件是并行工作的，在流水线执行单元并行执行各个命令操作的同时，命令解释器——从命令缓冲取出命令，进行预先处理，采用交叠式提交方法，每次向流水线提交一条无相关性的命令。

5 所谓交叠式提交方法，是指在某一命令退出流水线的最后一次循环中就插入新的命令，如图 5 所示，G 命令在 A 命令退出的最后一次循环插入，这样可以避免气泡的出现。为了实现上述方法，需在命令退出流水线的最后一次循环的前一个流水线周期尽早给出命令将退出的信号；还要保证新插入命令和将退出命令不具有命令相关性；另外，插入和退出的指令之间还可能会存在寄存器构成的命令现场（指现场寄存器）的冲突，需要加以解决。
10

命令解释器的组成结构如图 3 所示，包括命令缓冲读控制器 21、命令寄存器 22、操作数处理单元 23、流化线初始化寄存器 24 及控制自动机 25，下面结合该结构对本实施例命令解释器提交命令方法的流程进行说明，如图 4 所示，本实施例方法包括以下步骤：

15 命令解释器的自动机，控制命令缓冲读控制器从命令缓冲读取命令，存入命令寄存器，步骤 100；

自动机对命令进行译码，剔除命令编码不正确及携带不合理命令参数的非法命令，步骤 110；

20 自动机控制操作数处理单元根据命令的种类准备流水线各段的初始操作参数，存入流水线初始化寄存器，步骤 120；

判断流水线是否处于非满状态（即少于六条指令），如果是，执行步骤 140，否则（即包含六条指令），执行步骤 150，步骤 130；

将命令直接插入，并结束，步骤 140；

等待流水线提供的命令将退出的信号，步骤 150；

25 收到命令将退出的信号后，判断待插入的新命令和需要退出的老命令是否存在命令相关性，如果是，执行步骤 170，否则执行步骤 180，步骤 160；

等待老命令退出后再插入，并结束，步骤 170；

在老命令退出流水线的最后一次循环中，向流水线插入新命令，步骤 180。

30 在步骤 120 中的初始操作参数，如果是读写存储器命令，需要提供存储

器访问地址及必要的写入数据；如果是查找命令，需要提供查找关键字和采用的查找算法；如果是读写控制寄存器命令，需要提供寄存器地址及必要的写入数据。各类命令都还需要提供进入流水线的初始状态，即在加入流水线时，要将命令字映射到流水线的现场寄存器值。

5 步骤 150 中，为了保证命令解释及时决定命令插入的时机，需在命令退出流水线的最后一次循环的前一个流水线周期尽早给出命令将退出的信号，供命令解释器决定命令插入的时机。退出流水线的信号发出位置是固定的，本实施例是在新指令加入流水线段的前两级。以图 5 为例，新指令 G 在第 1 号流水线段插入，则老指令 A 退出流水线信号则在前一流水线周期的第 5 级 10 发出，在待退出指令 A 流经第 6 级流水线时，命令加入判断逻辑，决定是否加入新的指令 G。

在步骤 160 中，不能存在命令相关性是指需要退出的命令和插入的命令不能访问同一个部件，如插入指令和退出指令不能同时使用同一流水线段上的硬件处理模块部件，这可以由命令的操作流程预先确定。

15 在步骤 180 中，新命令的插入时，由于插入命令的现场在若干流水线级上依次加入供操作使用，而将退出的命令现场也可能在最后的若干级上被使用，就可能需要同时使用一个命令现场寄存器而构成冲突。哪些现场需要在哪些流水线段中使用是由各种命令的操作流程决定的，根据交叠命令序列可以预先确定新老现场的冲突。

20 在大多数的应用中，同一个现场刚好可以在插入命令需要使用前，而退出命令最后一次使用后被切换为新命令的现场，现场切换的硬件结构如图 6 所示，现场寄存器 61 经处理模块 62 连接到多路开关 63，新现场寄存器（新命令）64 则直接连接到多路开关 63，多路开关 63 的输出与下一级流水线的现场寄存器 65 相连。在正常情况下，上一级流水线的现场寄存器 61 内容经处理模块 62 处理进入下一级流水线的现场寄存器 65，而在命令交叠时，新现场寄存器 64 进入下一级流水线的现场寄存器 65，老命令的现场（位于上一级流水线的现场寄存器）则被丢弃。图 5 中产生命令交叠的三级流水线（流水线段 1, 2, 3）都可能产生上述现场寄存器的切换，在哪一级发生切换可以根据具体的操作命令在插入新命令的前一级进行判断。

30 在有些应用中，现场冲突无法避免，即新老命令还需要在同一流水线段

上使用现场寄存器。此时，产生一个现场分支，即新命令的现场加入到流水线中，老命令现场将进入现场分支，该分支将一直维持到老命令最后一次使用该现场进行操作为止。图 7 示出了一个流水线段的现场分支的硬件结构，其余段均与其相同。所谓现场分支是指有两路现场寄存器，一路作为主现场
5 寄存器 71，一路作为现场分支寄存器 72，它们通过一个多路开关 73 连接到处理模块 74 的输入上。

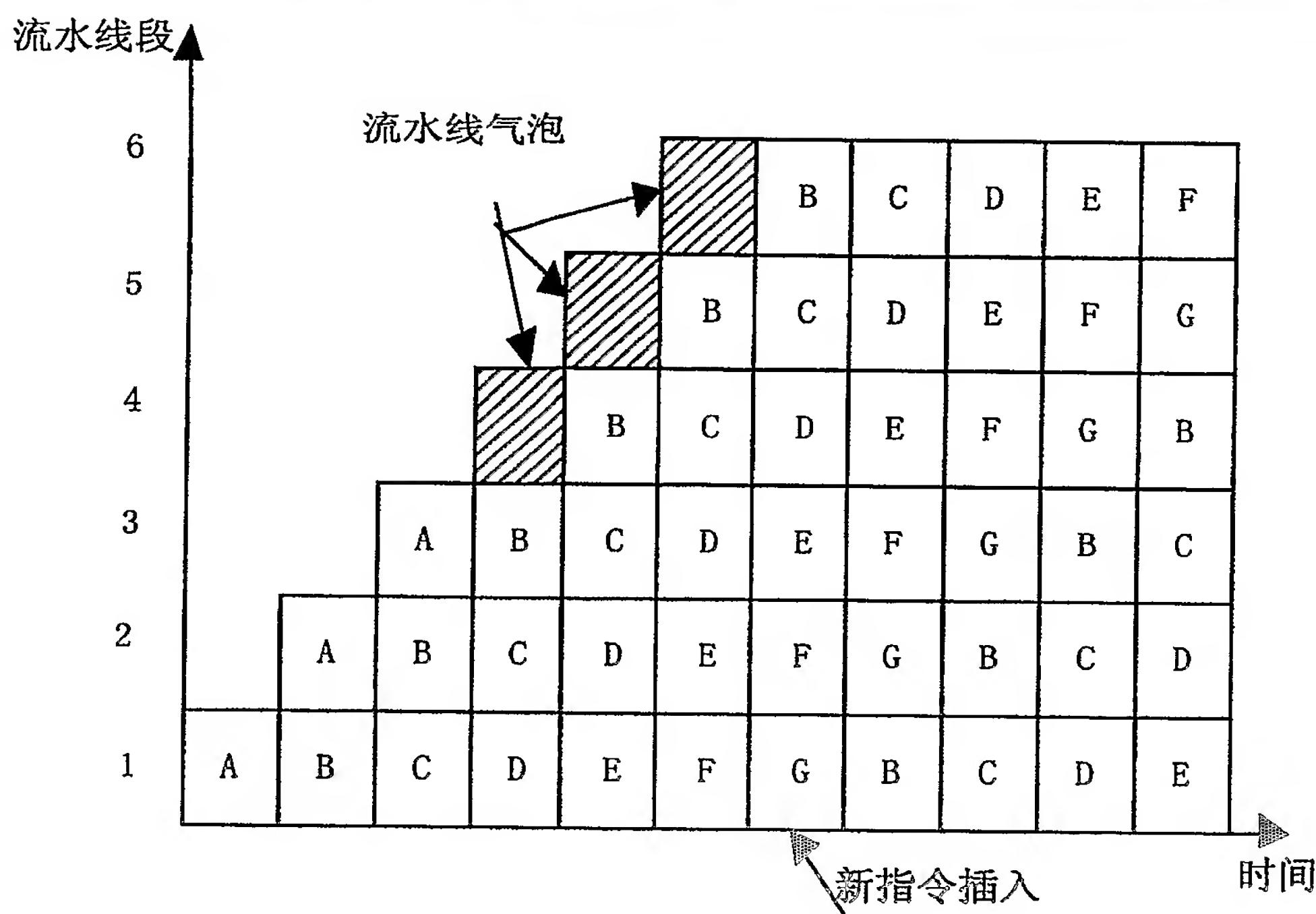
如前所述，可以交叠的命令是不会发生处理模块冲突的，如果新命令需要处理，处理模块接收主现场寄存器的输入，如果老命令需要处理，处理模块接收现场分支寄存器的输入，不需处理的寄存器可以直接进入下一段的同
10 类现场寄存器，在老命令最后一次使用该现场之后，再丢弃现场分支寄存器的值。对于现场冲突的判断，是在判断相关性的同时，即插入新命令的前一级进行的，老命令进入现场分支是从第一级进行。

请同时参照图 1 和图 5，在本发明的交叠式命令提交方法中，G 命令在 A 命令退出流水线的最后一次循环中插入，避免了“气泡”的出现，最多可
15 以比未改进的嵌入处理器命令提交方法提前 6 个时钟周期将后续命令插入到流水线中。命令提交的等待间隔也缩短了 6 个时钟周期。由此可以看出，本发明的新的命令插入方式——交叠式命令提交方法，可有效提高流水线各部件的利用率，进一步提高命令的执行速度。

虽然上述实施例是以一种特定的通信芯片为例，但很明显，本发明的动态循环流水线的交叠式命令提交方法，并不局限于处理某种或某几种命令，因而可以应用于其他具有环行流水线结构，且此流水线的命令加入和退出级
20 不是同一级的芯片中。

说 明 书 附 图

010300071



1

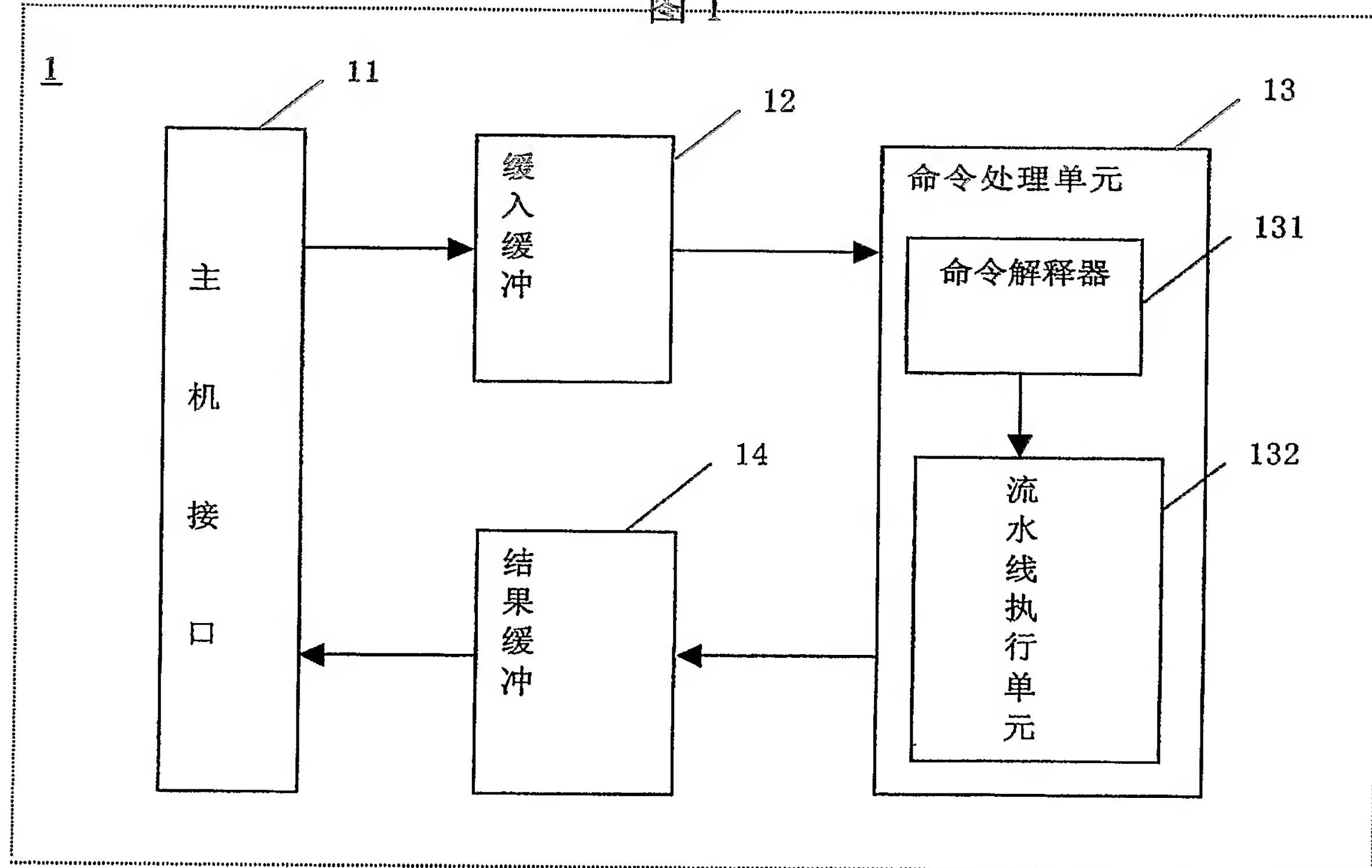


图 2

003-12-20

14

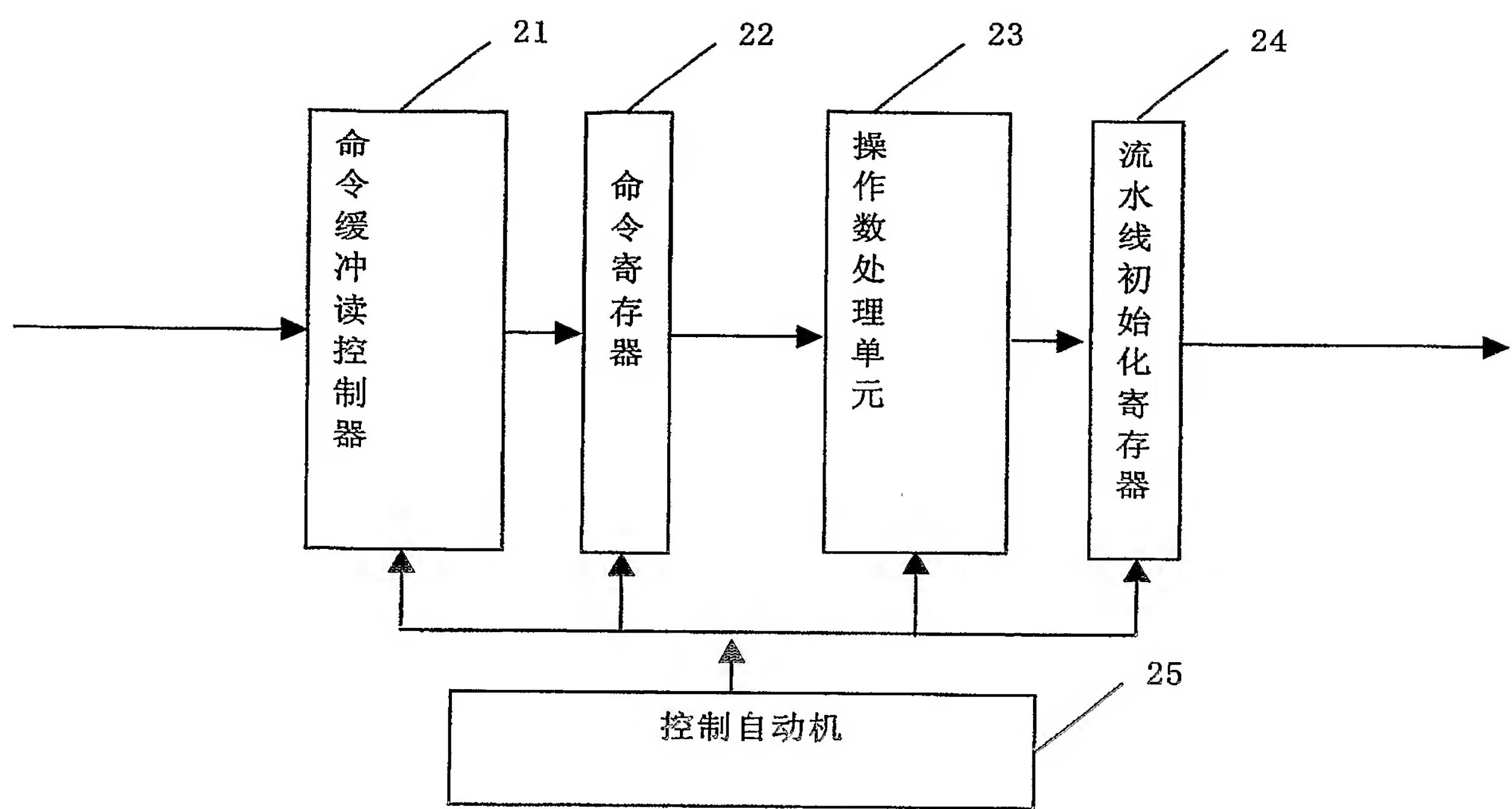


图 3

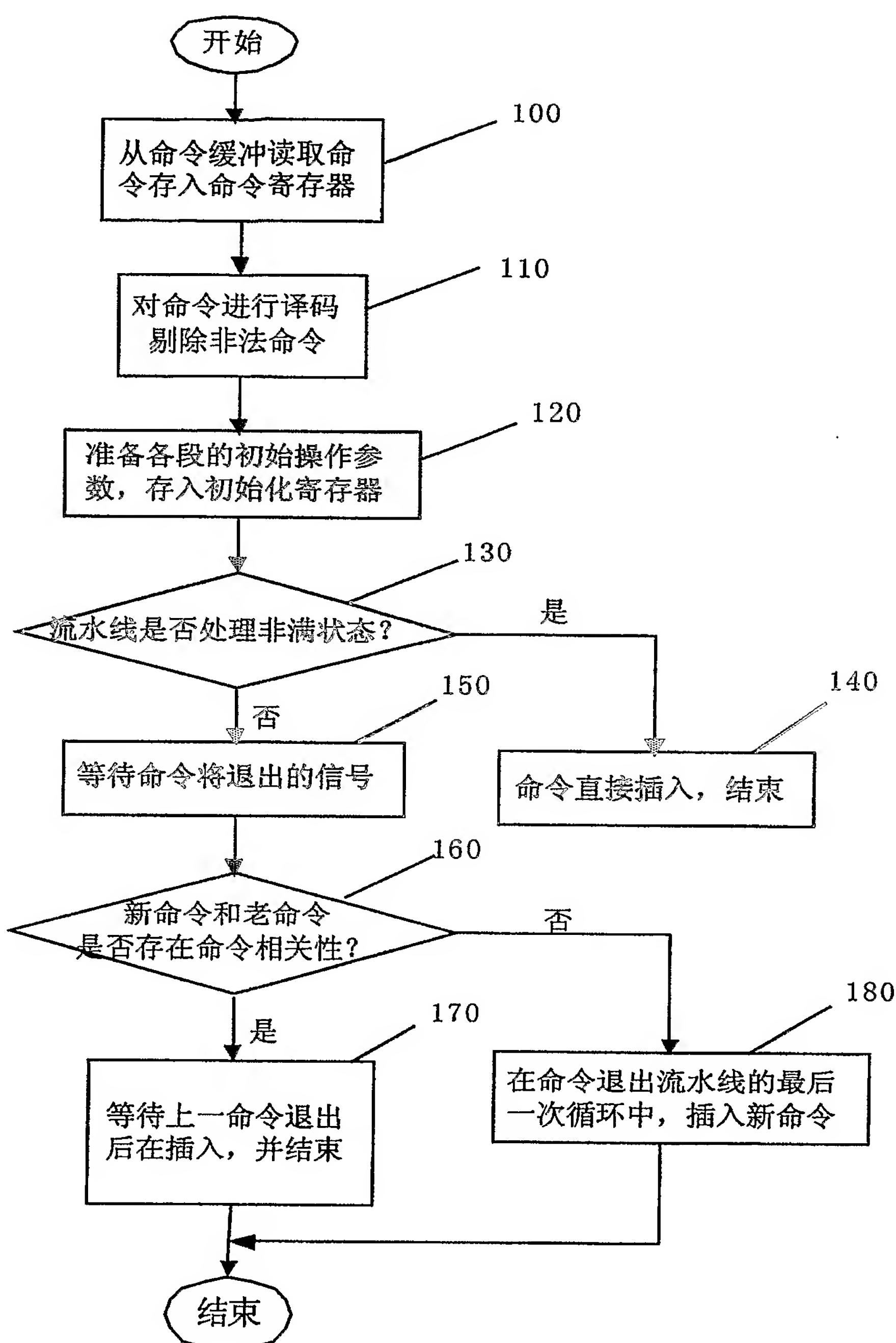


图 4

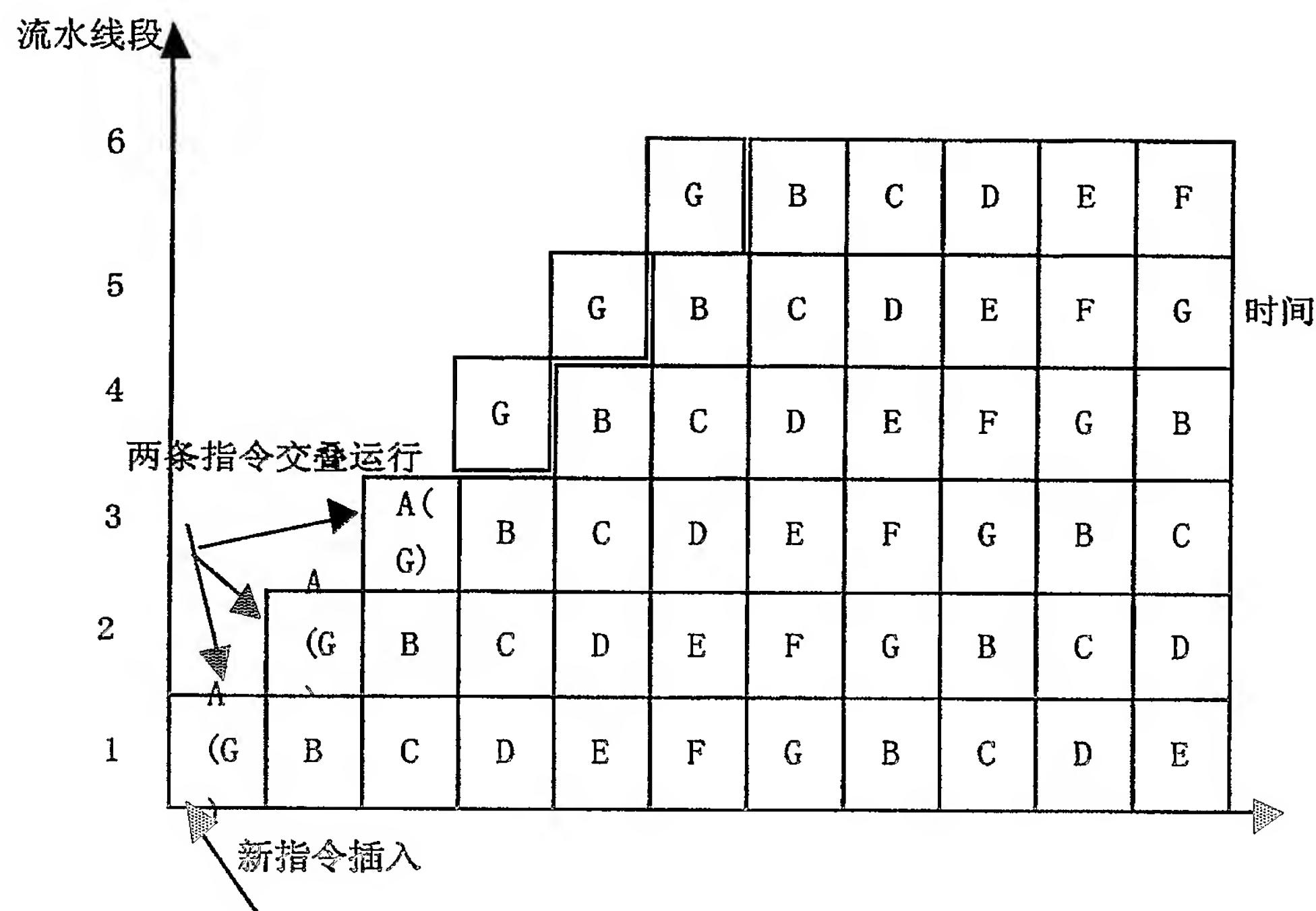


图 5

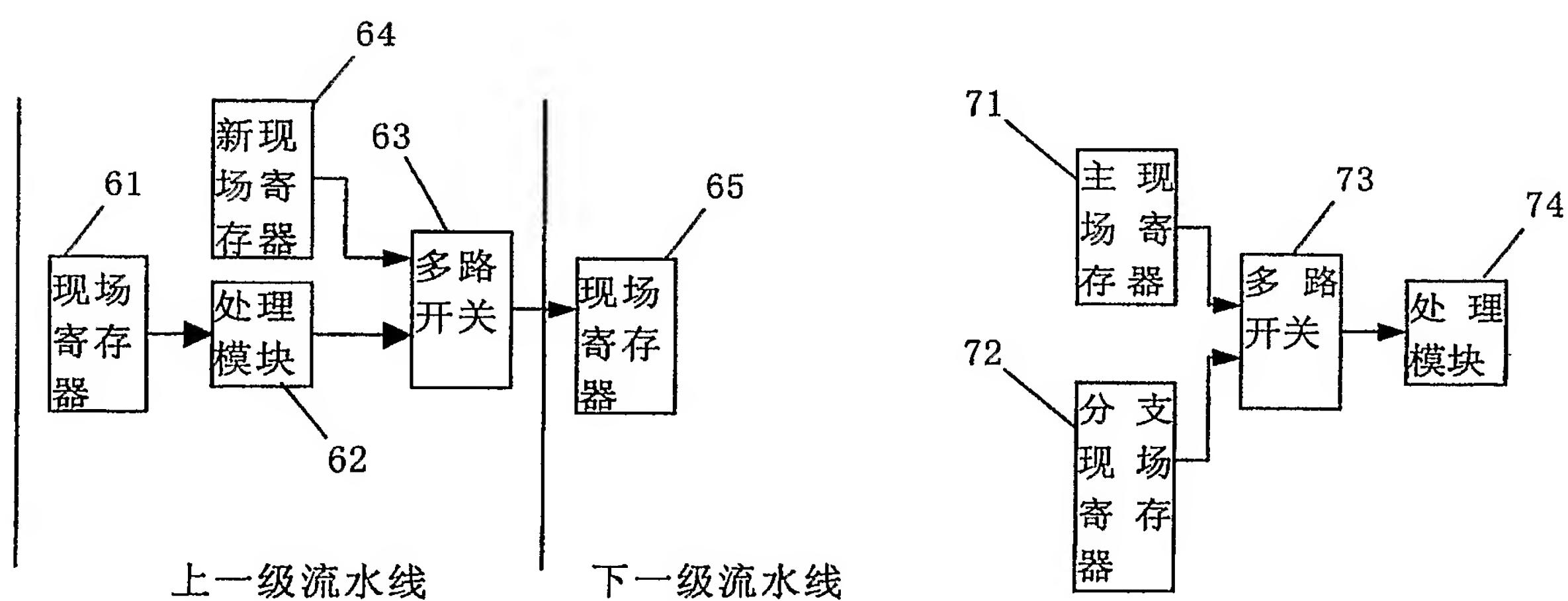


图 6

图 7